

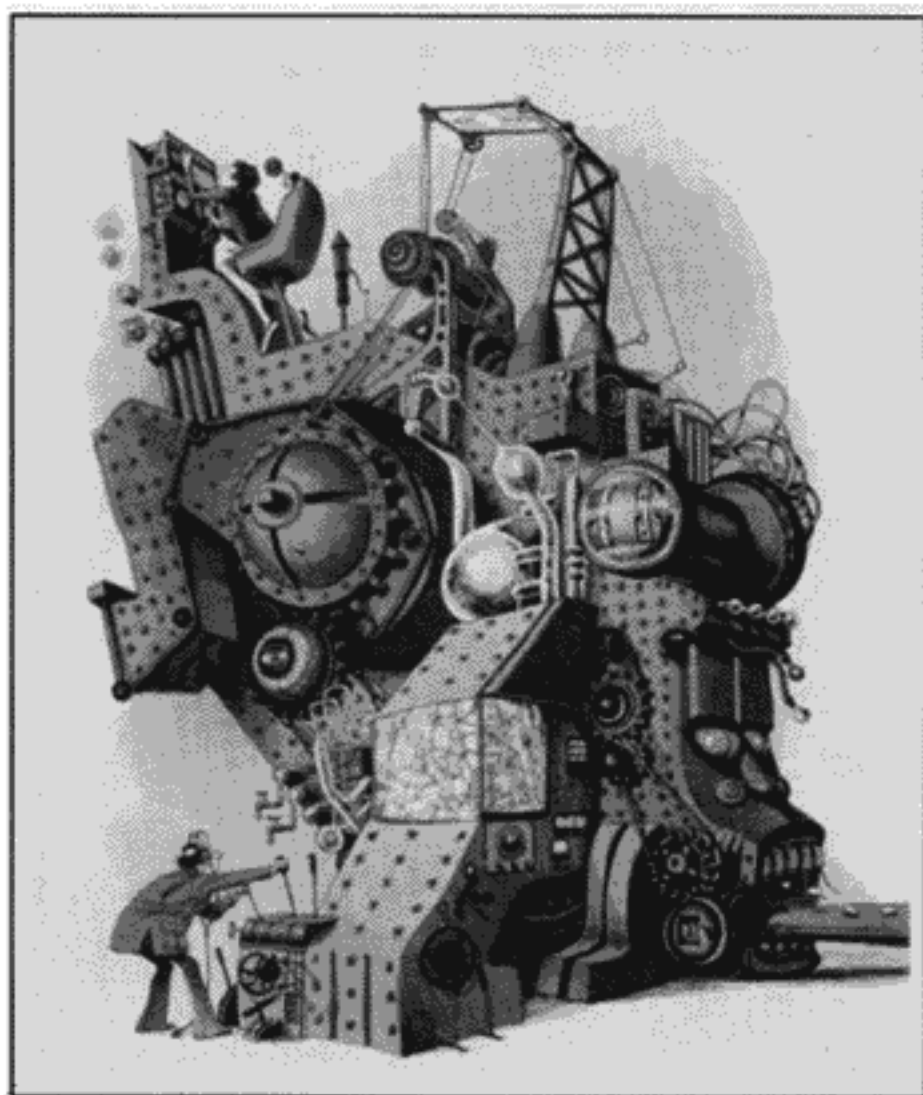
# El concepto de vectorización

CARLOS AMADOR

**E**l concepto de vectorización es, en realidad, muy sencillo. Pero, como pasa con la inmensa mayoría de los conceptos científicos, requiere de una larga introducción que le dé contexto. En contraste, el arte, la metafísica y unas cuantas ideas científicas (notables excepciones) no necesitan de esta introducción. Están cerca del hombre, son evidentes, tienen sentido inmediato, son universales. (Decimos que una idea científica es universal si, de acuerdo con la teoría aceptada en ese momento, se aplica o se cumple en todas las instancias que somos capaces de imaginar. Y no hacemos ninguna referencia a su alcance entre los hombres. Alternativamente, la otra universalidad se refiere precisamente a lo que nos alcanza a todos, a aquello a lo que todos somos sensibles. Una es una antropocentrismo en el clóset, la otra anda de loca.) Porque también es una imposición reciente la idea de que para apreciar una obra de arte se necesite una educación académica especial. Concedo la posibilidad de que haga falta una sensibilidad especial, un terreno fértil y receptivo. Pero esa fertilidad no requiere cursos, grados, problemas particulares específicos; para adquirirla basta con estar vivo; los únicos antecedentes necesarios son el amor, la pasión, el miedo, la soledad: algo de lo que todos sabemos.

En cambio, para entender un concepto científico, de los que podríamos llamar menores, y que son los más abundantes, hace falta haber viajado una larga ruta en un tren de pensamiento dado. Hace falta haberla seguido fielmente. No hay manera de llegar a ellos de golpe. En eso, algunos hemos encontrado motivo de orgullo y hemos llegado a suponer que esa particularidad hace de ésta la indiscutible forma superior de conocimiento, como si la universalidad de la otra fuera un defecto. La verdadera obra de arte, la intuición metafísica genial, es universal y evidente. Requiere de una mínima explicación, un esfuerzo accesible a todos los hombres. El concepto técnico, en cambio, implica necesariamente la prisión del contexto.

Así pasa con la vectorización. Una de las cosas más difíciles de enseñar en la física, en las matemáticas o en la química, es eso que en nuestro limitado vocabulario llamamos



respectivamente intuición física, matemática y química. Pedimos que, en el que aprende, se desarrolle pronto la capacidad de pensar en un problema como una situación natural. Decimos algo así como —Imagínate que vas por la calle y de pronto tienes que multiplicar una matriz por otra. Como cuando un maestro de cello le dice a su alumno —Así, natural. ¿Ves qué fácil?—, en el momento preciso en que esa posición le acalambra los brazos. ¿Por qué un ser humano ha de adoptar esa posición tan rara? ¿Por qué otro ha de multiplicar dos matrices? Porque esa es la ruta específica que les lleva a lo que se proponen. No hay otra; pero mientras, el primero produce algo que provocará una reacción en todos los hombres, el segundo produce solamente otra matriz (o, dependiendo de los detalles del problema, un vector e incluso un número).

Carlos Amador: Facultad de Química, UNAM.

Podemos construir un contexto para nuestra multiplicación. Podemos contar una larga historia que justifique ante nuestro auditorio, con más o menos éxito según convenza más o menos, la necesidad de multiplicar esas dos matrices. Podríamos, con alguna imaginación, encontrar un problema cuya solución requiera la dichosa multiplicación con el que nuestro auditorio alcance cierto grado de identificación. (Insisto, la limitación no está en nuestro auditorio.) A menudo, sin embargo, sucederá que en el camino, desde el problema planteado hasta la multiplicación, perdamos la atención de nuestro auditorio conforme lo vayamos elaborando. En esta ocasión ya nos arriesgamos bastante. Así que olvidemos el pretexto y empecemos a darle contexto al concepto de vectorización (que queremos demostrar que es muy sencillo), a partir de la concedida necesidad de multiplicar dos matrices.

Las matrices son conjuntos de números acomodados en renglones de acuerdo con una convención, por ejemplo:

$$\begin{array}{l} \text{renglón 1} \rightarrow \\ \text{renglón 2} \rightarrow \\ \text{renglón 3} \rightarrow \end{array} \begin{pmatrix} 10 & 7 & 2 \\ 1 & -3 & 5 \\ 2 & -4 & 0 \end{pmatrix}$$

Este acomodo particular sugiere otra descripción: las matrices son conjuntos de números acomodados en columnas. Por ejemplo:

$$\begin{array}{c} \text{columna} \\ 1 \\ 2 \\ 3 \end{array} \begin{pmatrix} 10 & 7 & 2 \\ 1 & -3 & 5 \\ 2 & 4 & 0 \end{pmatrix}$$

Es imposible vencer la tentación de hacer las definiciones que siguen. Así, decimos que la matriz de nuestro ejemplo es una matriz de tres renglones por tres columnas, o para ir más rápido, de 3 x 3. Y para referirnos a un elemento particular de esa matriz damos sus coordenadas; así, el siete que aparece en el primer renglón, en la segunda columna, es el elemento "12" (léase "uno dos" y no "doce").

Queremos multiplicar dos matrices, es decir, necesitamos otra. Sólo que en lugar de inventar otro conjunto de números cualesquiera para escribirla, podemos referirnos a cualquier matriz representando sus elementos en general. Supongamos que la nueva matriz está representada por:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{1}$$

Y ya que llegamos hasta aquí, olvidémonos de la primera matriz y utilicemos otra de estas generales:

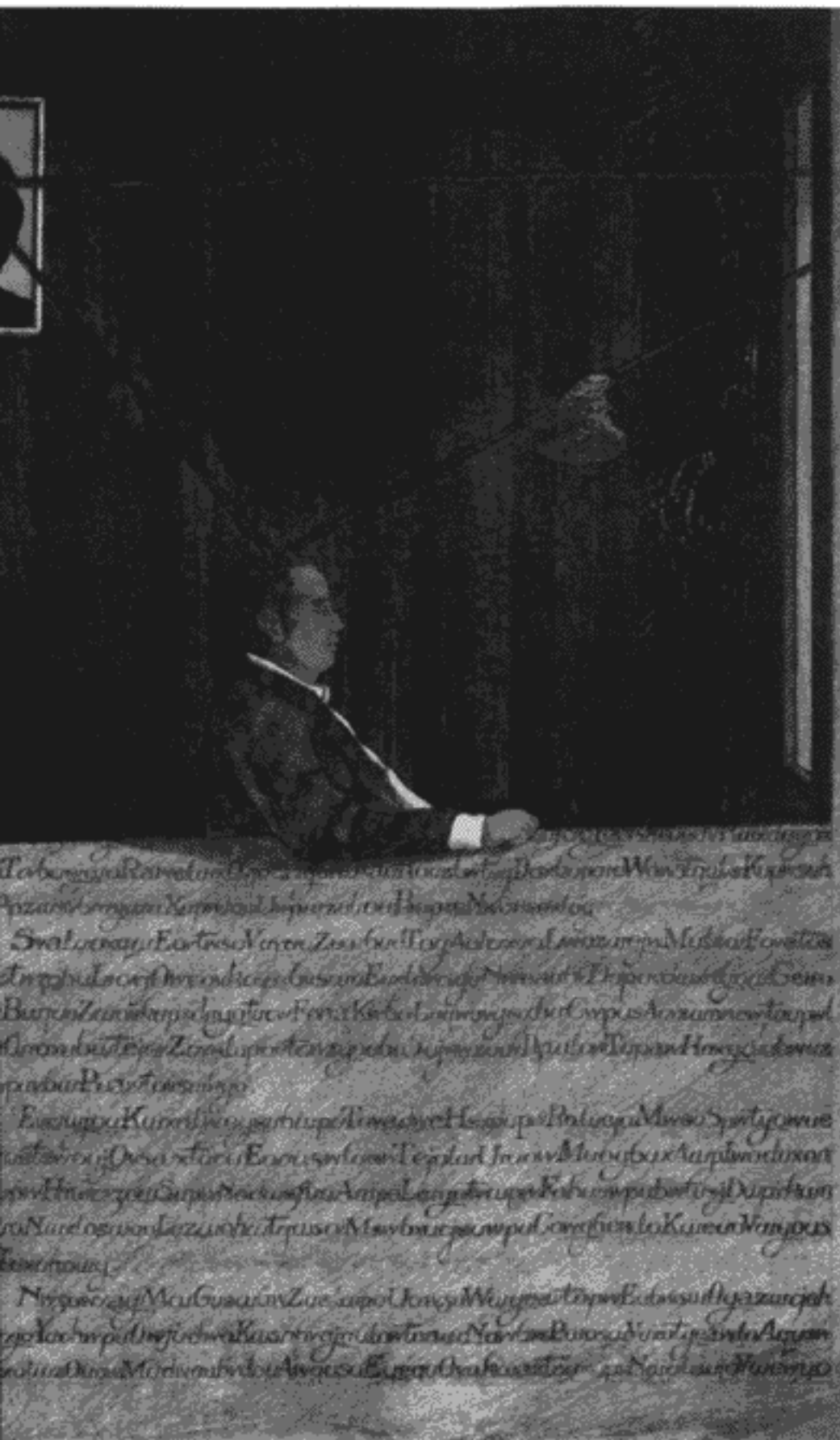
$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \tag{2}$$

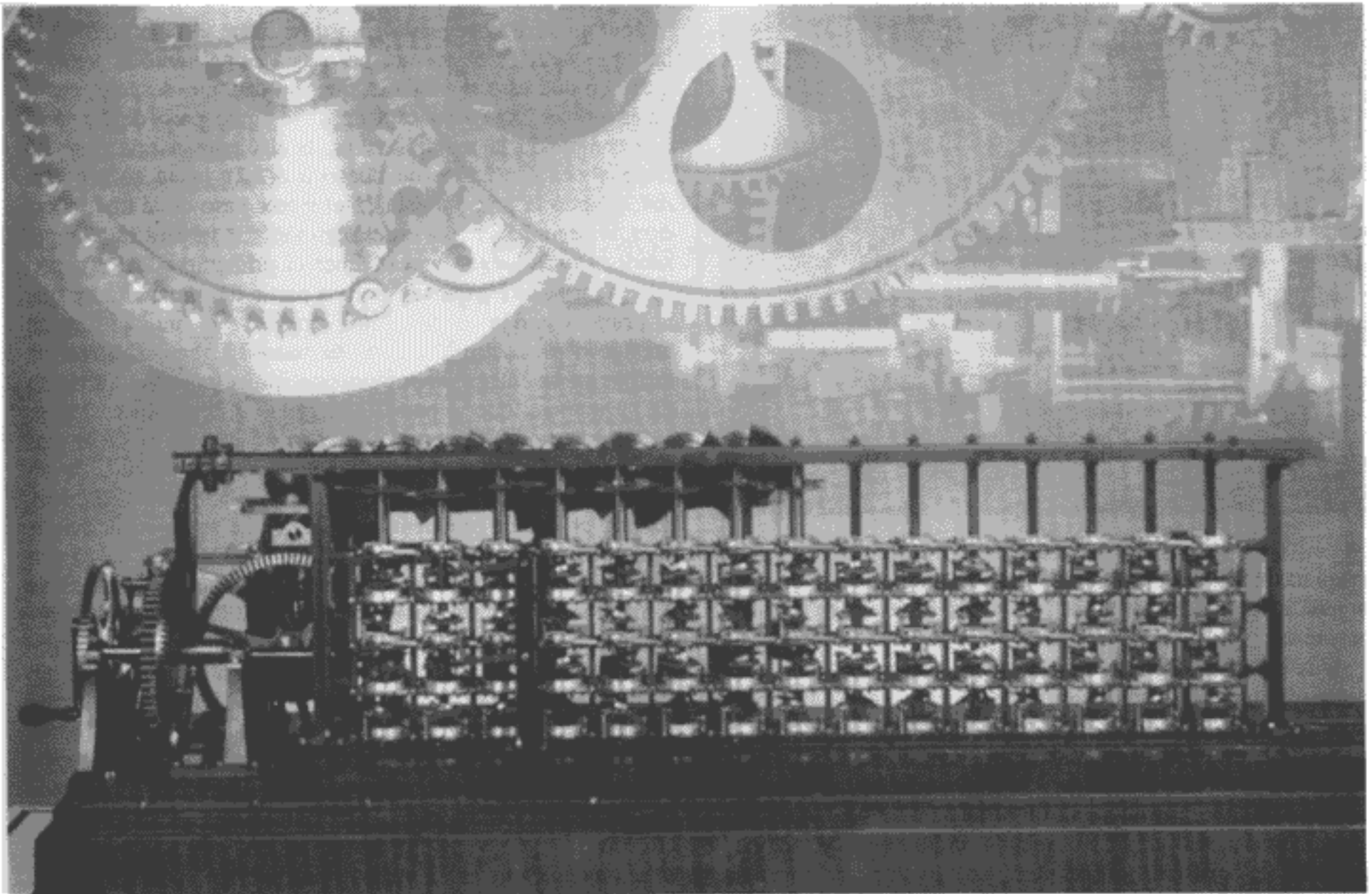
Ahora sí. Ya podemos hacer la multiplicación. Ésta se define (y aquí aparece de nuevo el fantasma que estamos tratando de exorcizar desde el principio; las preguntas ¿por qué se define así?, ¿para qué se define así?, son legítimas y deben contestarse, pero para llegar a demostrar que el concepto de vectorización es muy sencillo las tenemos que descartar por el momento) de la siguiente manera: el resultado de la multiplicación será otra matriz.

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

cuyos elementos son,

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j}, \tag{3}$$





(*i* y *j* representan, como convenimos, el renglón y la columna donde está situado el elemento  $c_{ij}$  de la matriz *C*). Existe una representación esquemática de la forma en que se obtiene cada elemento del producto:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Por ejemplo, para obtener el elemento  $c_{12}$  del resultado hay que utilizar los elementos del renglón 1, de la primera matriz y los de la columna 2 de la segunda. Y multiplicarlos entre sí —el primer elemento del renglón con el primero de la columna, el segundo con el segundo y así sucesivamente— y sumar los resultados. Este proceso es lo que está representado en la ecuación (4).

Ahora sí, ya casi llegamos a la vectorización. Sólo falta ver cómo se programa una multiplicación de matrices en una computadora. Queremos obtener todos los elementos de la matriz que resulta de multiplicar *A* por *B*. Es decir, queremos obtener los elementos de la matriz *C*,  $c_{ij}$  dados por la ecuación 4, para todos los valores de *i* y *j*. En nuestro ejemplo, tanto *i* como *j* pueden tomar los valores 1, 2 y 3. En FORTRAN esto se indica de la siguiente manera:

```
do 100 i = 1, 3
do 100 j = 1, 3
do 100 k = 1, 3
    c(i, j) = c(i, j) + a(i, k) * b(k, j)
100 continue.
```

La primera instrucción inicia un proceso repetitivo en el que el índice *i* va a tomar los valores, 1, 2 y 3. Para cada uno de estos valores existe otro proceso repetitivo en el que el índice *j* va a tomar los valores 1, 2 y 3. Finalmente, un último proceso repetitivo hace que *k* tome los valores 1, 2 y 3. Para cada terna de valores de *i*, *j* y *k*, la cuarta línea indica la operación que se efectúa entre los elementos correspondientes de las matrices y el elemento al cual se suma el resultado. El lector puede convencerse de que este proceso iterativo produce los elementos definidos por la ecuación (4).

Para cada valor de *i* (el ciclo repetitivo más interno), se suma a  $c(i, j)$  el número que resulta de la multiplicación de  $a(i, k) \cdot b(k, j)$ :  $a(i, k)$  es un número distinto para cada valor de *i* (que es el índice que está cambiando más aprisa), en cambio,  $b(k, j)$  es una constante independiente del valor (y del cambio en el valor) de *i*. Así lo que tenemos es la multiplicación de una variable,  $a(i, k)$ , por una constante,  $b(k, j)$ , y su adición a otra variable,  $c(i, j)$ .

Y, ahora sí, ya está todo listo para entender qué es la vectorización y por qué ésta hace tanto más rápidas a las supercomputadoras. Esencialmente, la diferencia está en la forma en que la computadora realiza el proceso indicado. Mientras que las



máquinas normales ejecutan sus instrucciones una por una, las supercomputadoras pueden hacer varias operaciones simultáneamente.

Una máquina normal (también llamadas secuenciales) funciona paso a paso. Cada determinado lapso, ejecuta una nueva instrucción. Y durante ese tiempo *sólo* hace eso. Como en el cine, la ilusión de continuidad proviene de la brevedad de ese lapso. (No son ya escasas las máquinas en las que ese lapso es de la cuarta parte de una millonésima de segundo.) La parte de la máquina donde se hacen las sumas y multiplicaciones se denomina procesador. En él están programados los algoritmos necesarios para hacer sumas, multiplicaciones, divisiones, comparaciones lógicas, y una o dos operaciones más. Supongamos que, en nuestro ejemplo, los índices  $k$  y  $j$  valen respectivamente 3 y 2, y nos encontramos al inicio del ciclo sobre  $i$ . El procesador trae de la memoria central el elemento  $b(3,2)$ , la siguiente instrucción (que se efectúa después de que ha transcurrido el lapso de tiempo mencionado) llama al elemento  $a(1,3)$ ; un instante después, ambos números entran en el algoritmo de la multiplicación. Éste consiste típicamente en unos siete pasos (comparar los signos, sumar los exponentes, etcétera) y, consume por lo tanto, siete instantes más. De nuevo, durante este tiempo la máquina no puede hacer ninguna otra cosa. Luego viene la suma; hay que traer el número  $c(1,2)$  y sumarlo al resultado de la multiplicación: otros siete u ocho instantes. En total, producir cada nuevo término consume unos veinte intervalos.

El procesador de una supercomputadora, en cambio, está diseñado para permitir cierto grado de simultaneidad. Los algoritmos que utiliza el procesador para sumar, multiplicar, dividir, etcétera, son idénticos a los de las otras máquinas: las diferencias se encuentran en la arquitectura misma de la máquina. La primera es que en el procesador de una supercomputadora pueden estar presentes, al mismo tiempo, varios números. Así, para hacer la multiplicación de nuestro ejemplo, la supercomputadora, en un solo lapso, lleva al procesador toda una columna de la matriz  $A$ :  $a(1,3)$ ,  $a(2,3)$  y  $a(3,3)$ . En seguida llama el número por el que tiene que multiplicar toda esta columna:  $b(3,2)$ . Con todos estos números en el procesador inicia las operaciones. Mete en el algoritmo de la multiplicación la primera pareja de números:  $a(1,3)$  y  $b(3,2)$ .

Y aquí aparece la segunda gran diferencia: el algoritmo de la multiplicación es el mismo, pero está *completamente segmentado*. Lo que esto significa es que, un instante después, cuando la primer pareja ya está en el segundo paso de los siete que componen la multiplicación, la siguiente pareja (que incluye ahora a  $a(2,3)$  y de nuevo a  $b(3,2)$ , inicia su proceso. Así, cuando la primera está en el tercer paso de la multiplicación, la segunda ya está en el segundo y la tercera está empezando. Desde luego, para obtener el primer producto, tienen que pasar varios instantes; pero a partir del momento en que la primera pareja acaba su proceso, cada instante se obtiene un nuevo producto. Y eso no es todo. La tercera diferencia es que en las supercomputadoras el procesador está dividido de tal manera, que la parte encargada de la multiplicación es independiente de la parte encargada de la suma (y de la parte que divide y de la que hace comparaciones lógicas), es decir, las *unidades funcionales son independientes*. Entonces, cuando el primer producto está disponible se puede iniciar la suma, sin esperar hasta que todos los productos se hayan calculado. La suma funciona de manera similar: las parejas de sumandos se van alimentando en el algoritmo una cada instante. Y, de nuevo, una vez que llenan esa "línea de montaje", es decir una vez que hay una pareja de sumandos en cada una de las etapas de la suma, cada instante sale un término; de manera muy similar a como se fabrican productos en serie: autos, televisores, computadoras (pero no supercomputadoras; de esas sólo se hacen unas cincuenta cada año).

Nuestro ejemplo es demasiado chico para llenar la línea de montaje. Pero es muy fácil imaginar la situación análoga en la que las matrices que multiplicamos tengan, digamos, cien renglones y cien columnas. Después del tiempo de "arranque" (el tiempo que tarda en llenarse la línea) la supercomputadora estará produciendo una multiplicación y una suma en cada instante. La aplicación de la vectorización aumenta, por sí sola, unas veinte veces la rapidez de cálculo. Además, ya que de todas maneras la supercomputadora va a ser una máquina extremadamente cara, los fabricantes se pueden dar el lujo de emplear los materiales más rápidos conocidos y que resultan demasiado costosos para las computadoras más modestas. Los instantes característicos de estas máquinas son de unas pocas mil millonésimas de segundo.

Desde luego, esperamos de una computadora que haga algo más que multiplicar matrices. Quizá resulte sorprendente saber que, sin embargo, la mayor parte de las aplicaciones científicas de las computadoras utilizan multiplicaciones de matrices, y, de hecho, consumen la mayor parte del tiempo en este simple proceso. Pero sí, es cierto: si un programa de computadora no multiplica matrices, el empleo de una supercomputadora es un poco excesivo. Excepto por otra característica fundamental de las supercomputadoras: algunas de ellas tienen varios procesadores trabajando simultáneamente. Todos ellos capaces de vectorización.

Y ya. Así de fácil. Desde luego, los problemas de diseño asociados con la aplicación de estas ideas tan simples son gigantescos. Y eso hace que cada supercomputadora cueste entre cinco y veinte millones de dólares y que los fabricantes en todo el mundo sólo sean cinco o seis. Pero, quisiera insistir, la idea de vectorización es sencilla. ♦